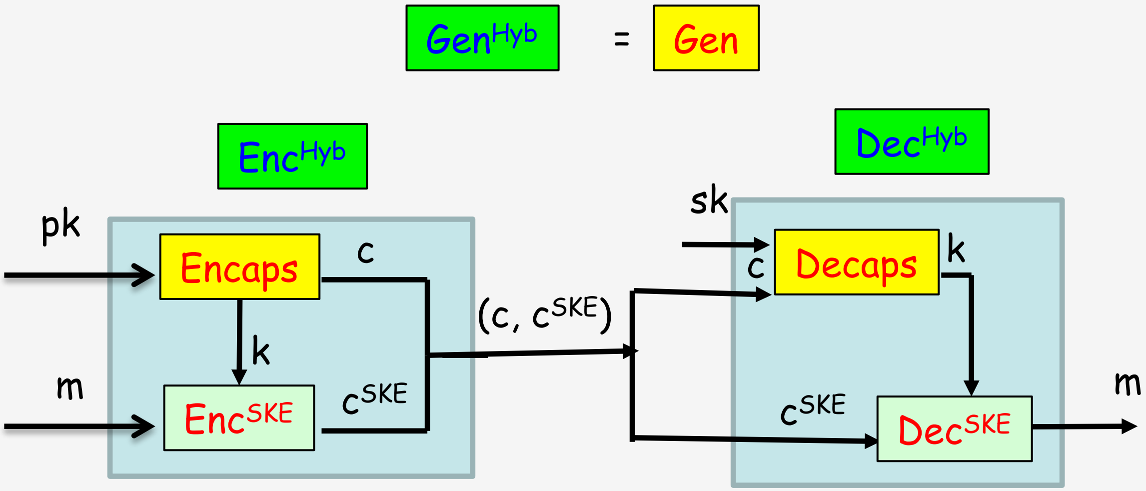
Cryptography

Lecture 11

Arpita Patra

Generic Results in PK World

CPA Security	CCA Security
Bit Encryption → Many-bit Encryption	Bit Encryption → Many-Bit Encryption
Π CPA-secure KEM Π^{SKE} COA-secure SKE → Π^{Hyb} CPA-secure	Π CCA-secure KEM Π^{SKE} CCA-secure SKE → Π^{Hyb} CCA-secure

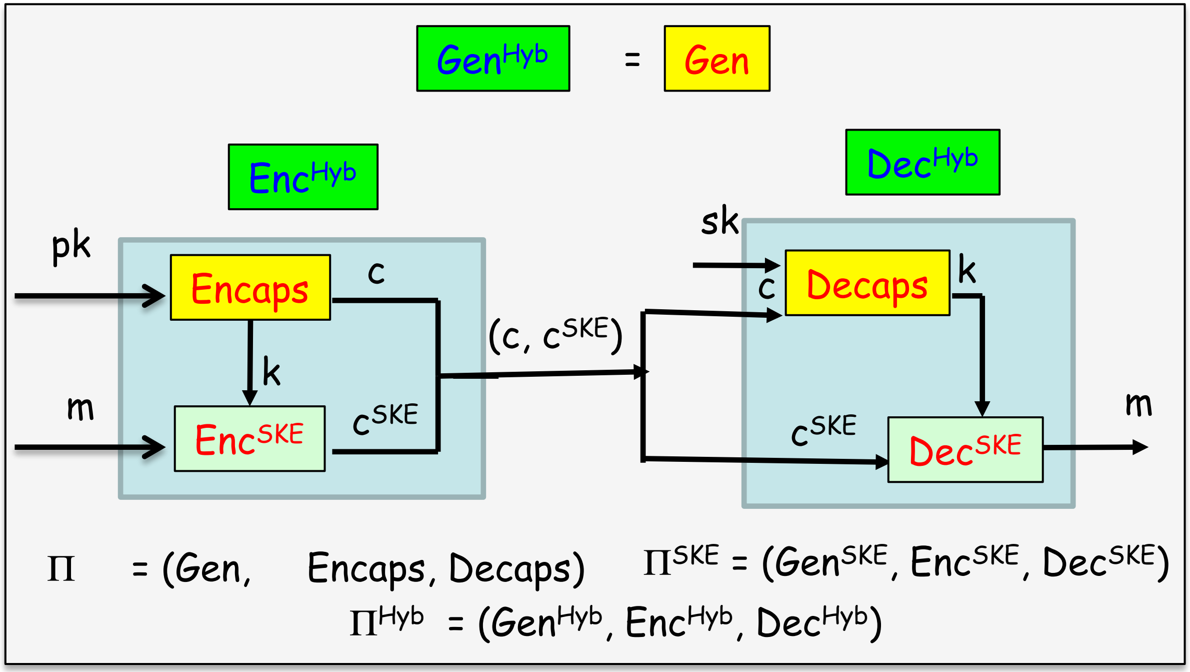


Π = (Gen, Encaps, Decaps) Π^{SKE} = (Gen^{SKE}, Enc^{SKE}, Dec^{SKE})

Π^{Hyb} = (Gen^{Hyb}, Enc^{Hyb}, Dec^{Hyb})

Constructions for PK World

CPA Security	**CCA Security**
PKE Instantiation: DDH based El Gamal	Instantiation: DDH + CR Hash based Cramer-Shoup Scheme (first ever CCA secure under standard assumption)
KEM Instantiation: HDH based variation of El Gamal	KEM Instantiation: ODH based (the same) variation of El Gamal
Variant El Gamal KEM PRG-based SKE → Π^{Hyb} CPA-secure	Variant ElGamal KEM CPA-secure SKE + sCMA MAC → Π^{Hyb} CCA-secure



Π = (Gen, Encaps, Decaps) Π^{SKE} = (Gen^{SKE}, Enc^{SKE}, Dec^{SKE})

Π^{Hyb} = (Gen^{Hyb}, Enc^{Hyb}, Dec^{Hyb})

CCA Security for KEM

CCA experiment $\text{KEM}^{\text{cca}}_{A,\Pi}(n)$

Also have to give $\text{Decaps}_{sk}(.)$ service to the attacker

$_{pk}(1^n)$

- k if b=0

← uniform random string, b =1

PPT A

I can break Π

pk

(c,k')

b' ∈ {0, 1}

(Attacker's guess about encapsulated key)

b ← {0, 1}

Let me verify $\xleftarrow{\text{pk, sk}}$ Gen(1^n)

Game Output

b = b' → 1 --- attacker won

b ≠ b' → 0 --- attacker lost

Π is CPA-secure if for every PPT attacker A, the probability that A wins the experiment is at most negligibly better than ½

$$\Pr\left(\text{KEM}^{\text{cca}}_{A,\Pi}(n) = 1\right) \leq \frac{1}{2} + \text{negl}(n)$$

El Gamal like KEM

Gen(1^n)

(G, o, q, g)

$h = g^{x}$. For random x

pk= (G,o,q,g,h), sk = x

Enc$_{pk}$(m)

$c_1 = g^y$ for random y

$c_2 = h^{y} . m$

c= (c_1,c_2)

- Need to choose m randomly
- Multiplication
- Ciphertext= 2 elements

Dec$_{sk}$(c)

$c_2 / (c_1)^x = c_2 . [(c_1)^x]^{-1}$

- Multiplication

Security: DDH Assumption

Gen(1^n)

(G, o, q, g)

$h = g^{x}$. For random x

pk= (G,o,q,g,h,H), sk = x

Encaps$_{pk}$(1^n)

$c = g^y$ for random y

$k = H(h^y) = H(g^{xy})$

(c,k)

- No need of that
- No Multiplication, hashing
- Ciphertext= 1 element

Decaps$_{sk}$(c)

$k = H(c^x) = H(g^{xy})$

- No Multiplication, hashing

Security: ??

El Gamal like KEM

Gen(1^n)

(G, o, q, g)

$h = g^{x}$. For random x

pk= (G,o,q,g,h,H), sk = x

Encaps$_{pk}$(1^n)

$c = g^y$ for random y

$k = H(h^y) = H(g^{xy.})$

(c,k)

Dec$_{sk}$(c)

$k = H(c^x) = H(g^{xy})$

ODH (Oracle Diffie-Hellman) Assumption

ODH problem is hard relative to (G, o) and hash function H: G -> $\{0,1\}^m$ if for every PPT A, (it is hard to distinguish $H(g^{xy})$ from a random string $\{0,1\}^m$ even given g^x, g^y AND an oracle $O_y(X)$: = $H(X^y)$; anything other than g^x can be quired)):

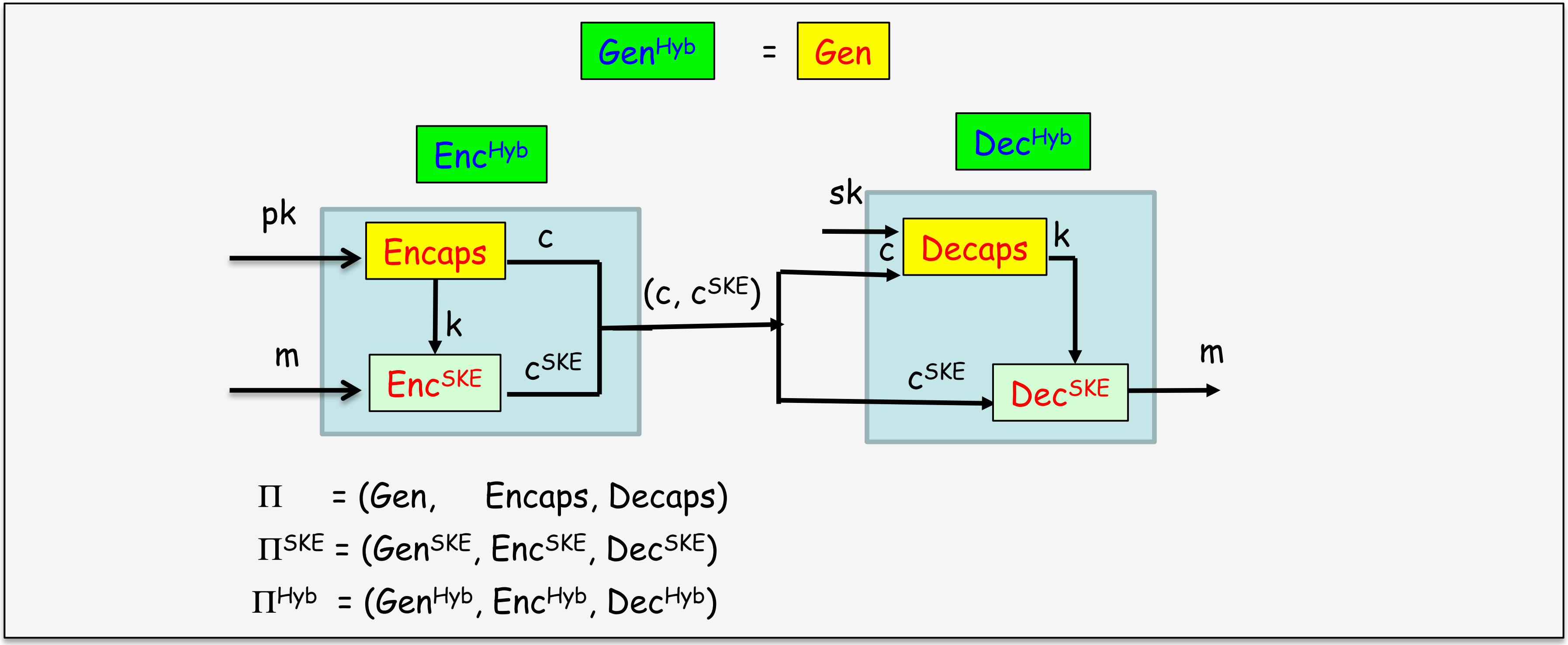
$$\left| \Pr[A^{O_y(.)}(G, o, q, g, g^x, g^y, H(g^{xy})) = 1] - \Pr[A^{O_y(.)}(G, o, q, g, g^x, g^y, r) = 1] \right| \leq \text{negl}()$$

ODH assumption is just the belief that there exist a group and hash function H so that the above is true.

It is stronger than HDH.

Theorem: ODH assumption holds → Π is a CCA-secure KEM

Construction of Hybrid CCA-secure PKE



Π = (Gen, Encaps, Decaps)

$\Pi^{\text{SKE}} = (\text{Gen}^{\text{SKE}}, \text{Enc}^{\text{SKE}}, \text{Dec}^{\text{SKE}})$

$\Pi^{\text{Hyb}} = (\text{Gen}^{\text{Hyb}}, \text{Enc}^{\text{Hyb}}, \text{Dec}^{\text{Hyb}})$

CCA Secure Π^{SKE} - CPA Secure Π^{SKE} + Strong CMA Secure Π^{MAC}

CCA Secure Π - Oracle Function assumption (ODH)

DHIES (Diffie-Hellman Integrated Encryption Scheme)- – ISO/IEC 18033-2

DHIES- - ISO/IEC 18033-2

Π (CCA) = (Gen, Encaps, Decaps)

Π^{SKE} (CPA) = (Gen^{SKE}, Enc^{SKE}, Dec^{SKE})

Π^{MAC} (sCMA)= (Gen^{MAC}, Mac, Vrfy)

Π^{Hyb} (CCA) = (Gen^{Hyb}, Enc^{Hyb}, Dec^{Hyb})

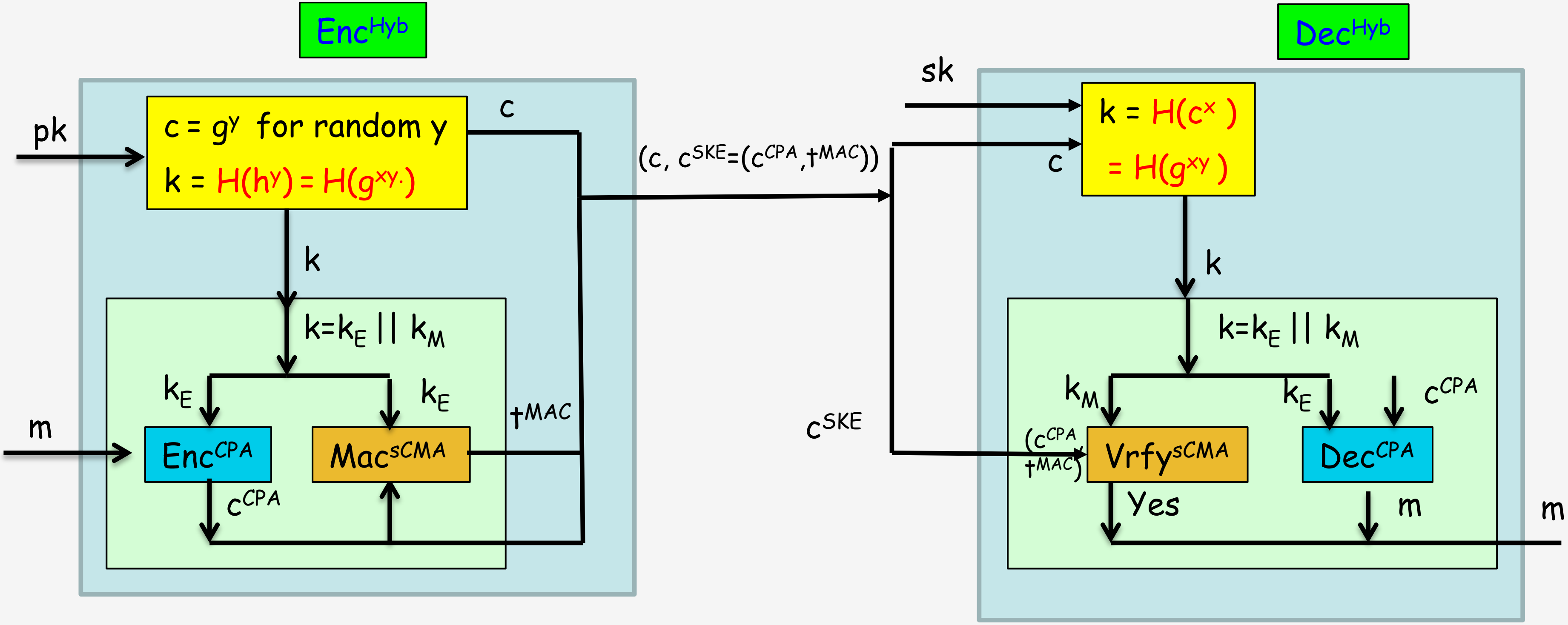
Gen^{Hyb} =

(G, o, q, g)

$h = g^{x.}$ For random x

H: G → $\{0,1\}^{2n}$

pk= (G,o,q,g,h,H), sk = x



CCA Secure Π^{SKE} - CPA Secure Π^{SKE} + Strong CMA Secure Π^{MAC}

CCA Secure Π - Number theoretic assumption + Hash Function (ODH)

DHIES (Term Paper)

Michel Abdalla, Mihir Bellare, Phillip Rogaway: **The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES.** CT-RSA 2001: 143-158

Cramer-Shoup Cryptosystem

Ronald Cramer, Victor Shoup:
A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack.
CRYPTO 1998: 13-25

Cramer-Shoup Cryptosystem- Route map

Another Look at DDH Assumption/ An alternative Formulation

↓

CPA Secure Scheme (different from El Gamal)

↓

CCA1 Secure Scheme

↓ + Collision-Resistant Hash Function

CCA Secure Scheme

Another Look at DDH

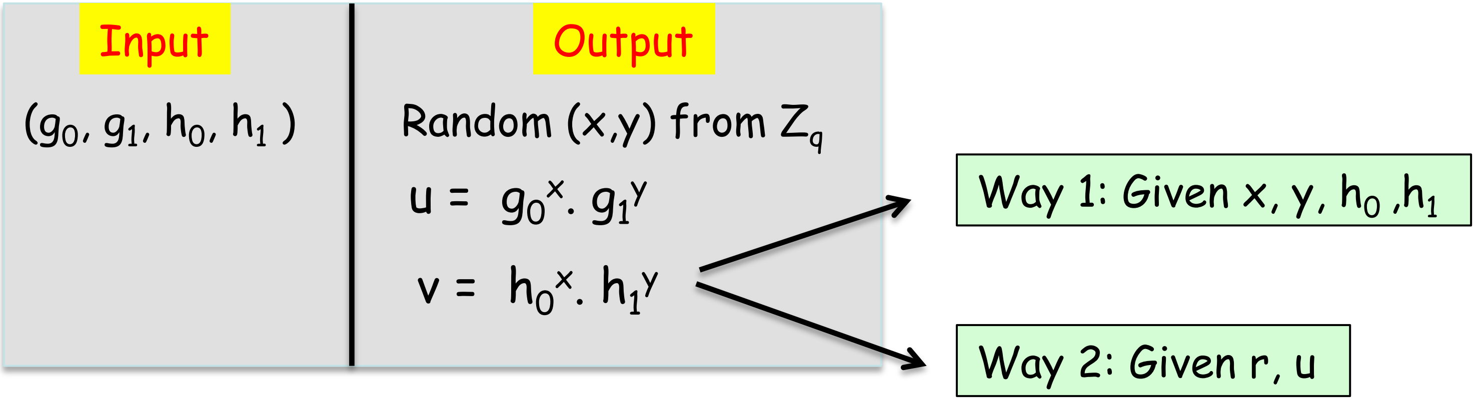
(G,o,q,g) – Group of Prime Order q

$$\left| \Pr[A(g, g^x, g^y, g^{xy}) = 1] \quad - \quad \Pr[A(g, g^x, g^y, g^z) = 1] \right| \leq \text{negl}()$$

(g_0, g_1, g_0^y, g_1^y) $\qquad$ $(g_0, g_1, g_0^y, g_1^{y'})$

$$\left| \Pr[A(g_0, g_1, g_0^r, g_1^r) = 1] \quad - \quad \Pr[A(g_0, g_1, g_0^r, g_1^{r'}) = 1] \right| \leq \text{negl}()$$

Randomization Function



Input

(g_0, g_1, h_0, h_1)

Output

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$v = h_0^x . h_1^y$

Way 1: Given x, y, h_0, h_1

Way 2: Given r, u

Case I $(g_0, g_1, h_0 = g_0^r, h_1 = g_1^r)$:

Claim: $u^r = v$

Proof: $u^r = (g_0^x . g_1^y)^r = h_0^x . h_1^y = v$

Case II $(g_0, g_1, h_0 = g_0^r, h_1 = g_1^{r'})$:

Claim: An all powerful adv A can guess v with probability at most $1/|G|$, even given (g_0, g_1, h_0, h_1, u).

Proof: A can compute r, r', α (where $g_1 = g_0^\alpha$) and discrete log of u (say R)

$$u = g_0^R = (g_0^x . g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R \text{ --- (1)}$$

$$v = h_0^x . h_1^y = g_0^{rx} . g_1^{r'y} = g_0^{rx + \alpha r'y}$$

$rx + \alpha r'y$ is linearly independent of $x + \alpha y$

For every guess R′ of this value $rx + \alpha r'y$, there exist a pair of unique values for x,y satisfying equation (1)

CPA Secure Scheme

$\mathrm{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

pk= (G,o,q, g_0,g_1, u), sk = (x,y)

$\mathrm{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1 , c)

$\mathrm{Dec}_{sk = (x,y)}(h_0, h_1 , c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

A, p(n): $\Pr\left(\mathrm{PubK}^{cpa}_{A,\Pi}(n) = 1 \right) > \frac{1}{2} + 1/p(n)$

DDH or non-DDH tuple?

$(G,o,q,g_0, g_1, h_0, h_1) \longrightarrow$

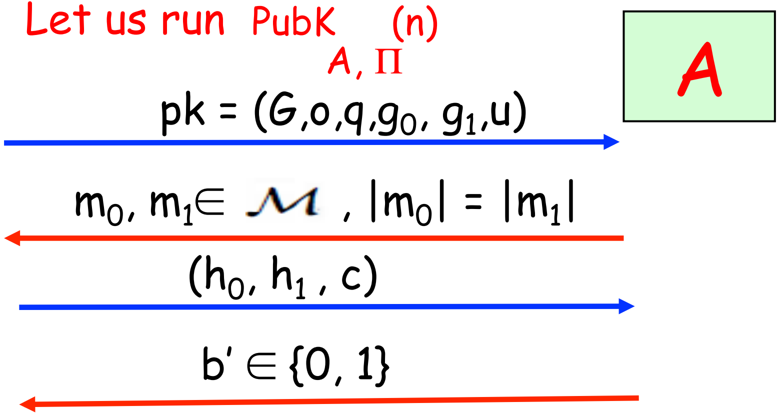
D

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$c = h_0^x . h_1^y . m_b$

Let us run $\mathrm{PubK}^{cpa}_{A,\Pi}(n)$



A

pk = (G,o,q,g_0, g_1,u) →

← $m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h_0, h_1 , c) →

← $b' \in \{0, 1\}$

CPA Secure Scheme

$\mathrm{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G, o, q, g_0, g_1, u)$, $sk = (x,y)$

$\mathrm{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1, c)

$\mathrm{Dec}_{sk = (x,y)}(h_0, h_1, c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

$$A, p(n): \quad \Pr\left(\mathrm{PubK}^{cpa}_{A,\Pi}(n) = 1 \right) > \tfrac{1}{2} + 1/p(n)$$

DDH Tuple

$(G, o, q, g_0, g_1, h_0 = g_0^r, h_1 = g_1^r)$ $\longrightarrow$

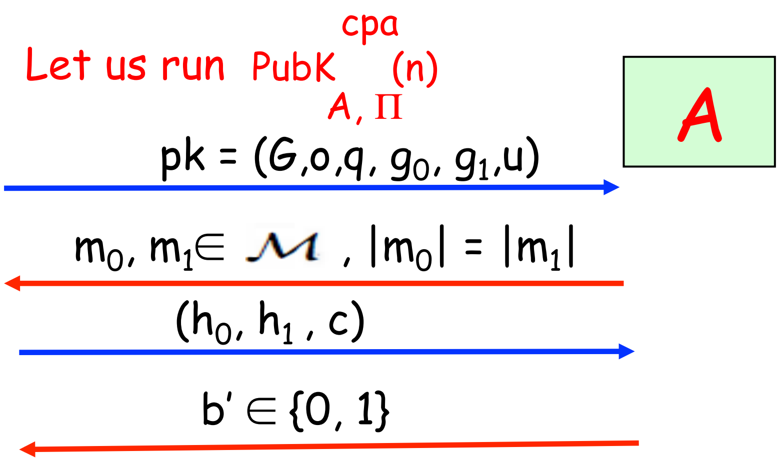
D

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$h_0^x . h_1^y = g_0^{rx} . g_1^{ry} = u^r$

$c = h_0^x . h_1^y . m_b$



Let us run $\mathrm{PubK}^{cpa}_{A,\Pi}(n)$

A

$pk = (G, o, q, g_0, g_1, u)$

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h_0, h_1, c)

$b' \in \{0, 1\}$

CPA Secure Scheme

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

pk= (G,o,q, g_0,g_1, u), sk = (x,y)

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1, c)

Dec$_{sk = (x,y)}$$(h_0, h_1, c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

A, p(n): $\Pr\left(\mathrm{PubK}^{cpa}_{A,\Pi}(n) = 1 \right) > \frac{1}{2} + 1/p(n)$

$\Pr\left(\mathrm{PubK}^{cpa}_{A,\overline{\Pi}}(n) = 1 \right) = \frac{1}{2}$

Non-DDH Tuple

$(G,o,q,g_0, g_1, h_0 = g_0^r, h_1 = g_1^{r'}) \longrightarrow$

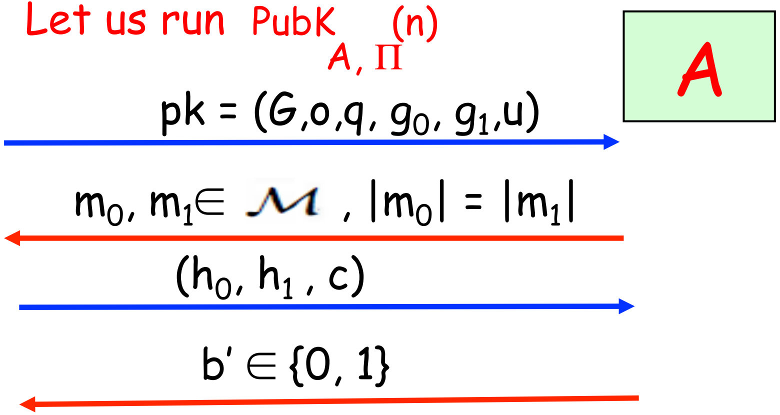
D

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$h_0^x . h_1^y$ is uniformly random element

$c = h_0^x . h_1^y . m_b$



Let us run $\mathrm{PubK}^{cpa}_{A,\Pi}(n)$

$pk = (G,o,q, g_0, g_1,u) \longrightarrow$

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1| \longleftarrow$

$(h_0, h_1, c) \longrightarrow$

$b' \in \{0, 1\} \longleftarrow$

CPA Secure Scheme

$\mathrm{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G, o, q, g_0, g_1, u)$, $sk = (x,y)$

$\mathrm{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1, c)

$\mathrm{Dec}_{sk = (x,y)}(h_0, h_1, c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

A, p(n):

$$\begin{array}{ccc} \Pr\left(\mathrm{PubK}^{\mathrm{cpa}}_{A,\Pi}(n) = 1\right) > \frac{1}{2} + 1/p(n) & & \Pr\left(\mathrm{PubK}^{\mathrm{cpa}}_{A,\overline{\Pi}}(n) = 1\right) = \frac{1}{2} \\ = & & = \\ \Big| \Pr[D(\text{DDH tuple}) = 1] & - & \Pr[D(\text{non-DDH tuple}) = 1] \Big| > 1/p(n) \end{array}$$

DDH or non-DDH tuple?

$(G, o, q, g_0, g_1, h_0, h_1)$ →

1 if b = b'

0 otherwise ←

D

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$c = h_0^x . h_1^y . m_b$

Let us run $\mathrm{PubK}^{\mathrm{cpa}}_{A,\Pi}(n)$

A

$pk = (G, o, q, g_0, g_1, u)$ →

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$ ←

(h_0, h_1, c) →

$b' \in \{0, 1\}$ ←

Why NOT El Gamal?

Gen(1^n)

(G, o, q, g)

Random x from Z_q, h = g^x

pk= (G,o,q, g_0, h), sk = x

$Enc_{pk}(m)$

$c_1 = g^y$ for random y

$c_2 = g^{xy}\cdot. m$

$Dec_{sk}(c)$

$c_2 / (c_1)^x = c_2 . [(c_1)^x]^{-1}$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

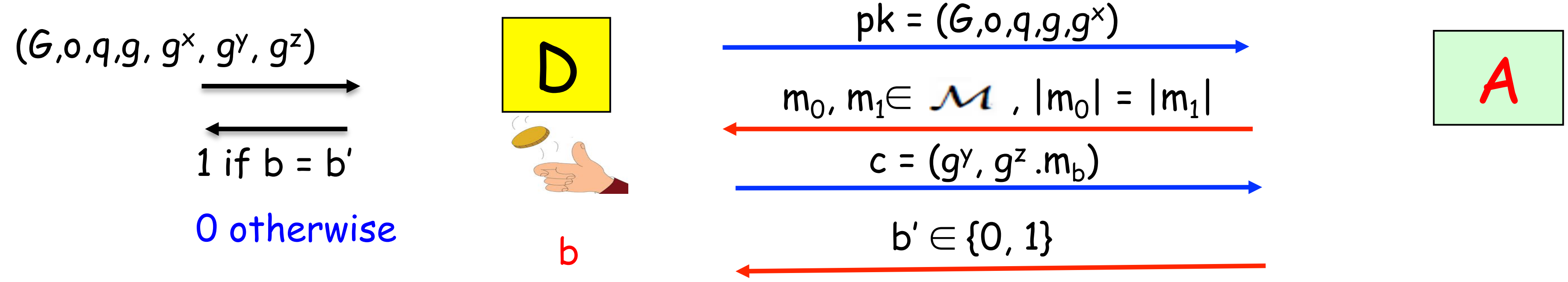
A, p(n): $\Pr\left(\mathrm{PubK}^{cpa}_{A,\Pi}(n) = 1\right) > \frac{1}{2} + 1/p(n)$ $\qquad \Pr\left(\mathrm{PubK}^{cpa}_{A,\overline{\Pi}}(n) = 1\right) = \frac{1}{2}$

$$\left| \Pr[D(\text{DDH tuple}) = 1] - \Pr[D(\text{non-DDH tuple}) = 1] \right| > 1/p(n)$$

The secret key is not with the reduction and so cannot provide DO service to A!

DDH or non-DDH t

us run $\mathrm{PubK}^{cpa}_{A,\Pi}(n)$



(G,o,q,g, g^x, g^y, g^z)

1 if b = b'

0 otherwise

D

b

pk = (G,o,q,g,g^x)

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

c = $(g^y, g^z .m_b)$

b' ∈ {0, 1}

A

Is the Scheme CCA secure?

Gen(1^n)

(G, o, q, g_0,g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

pk= (G,o,q, g_0,g_1, u), sk = (x,y)

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1 , c)

Dec$_{sk = (x,y)}$(h_0, h_1 , c)

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

It is malleable. Not CCA Secure

Is the Scheme CCA1 secure?

$\mathrm{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G, o, q, g_0, g_1, u)$, $sk = (x,y)$

$\mathrm{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1 , c)

$\mathrm{Dec}_{sk = (x,y)}(h_0, h_1 , c)$

$v = h_0^x . h_1^y$ (Way1)

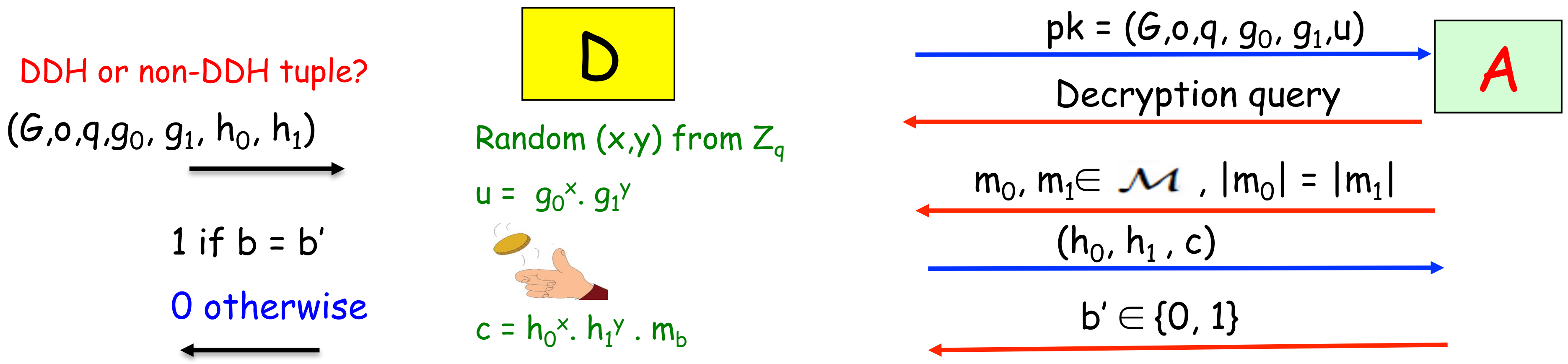
$m = c/v$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

A, p(n): $\Pr\left(\mathrm{PubK}^{cpa}_{A,\Pi}(n) = 1\right) > \frac{1}{2} + 1/p(n)$ $\Pr\left(\mathrm{PubK}^{cpa}_{A_u,\overline{\Pi}}(n) = 1\right) = \frac{1}{2}$

$$\left|\ \overset{=}{\Pr\,[D(\text{DDH tuple}) = 1]} - \overset{=}{\Pr\,[D(\text{non-DDH tuple}) = 1]}\ \right| > 1/p(n)$$



DDH or non-DDH tuple?

$(G, o, q, g_0, g_1, h_0, h_1)$

1 if $b = b'$

0 otherwise

D

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$c = h_0^x . h_1^y . m_b$

$pk = (G, o, q, g_0, g_1, u)$

Decryption query

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h_0, h_1 , c)

$b' \in \{0, 1\}$

A

Is the Scheme CCA1 secure?

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G, o, q, g_0, g_1, u)$, $sk = (x,y)$

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1, c)

Dec$_{sk = (x,y)}(h_0, h_1, c)$

$v = h_0^x . h_1^y$ (Way1)

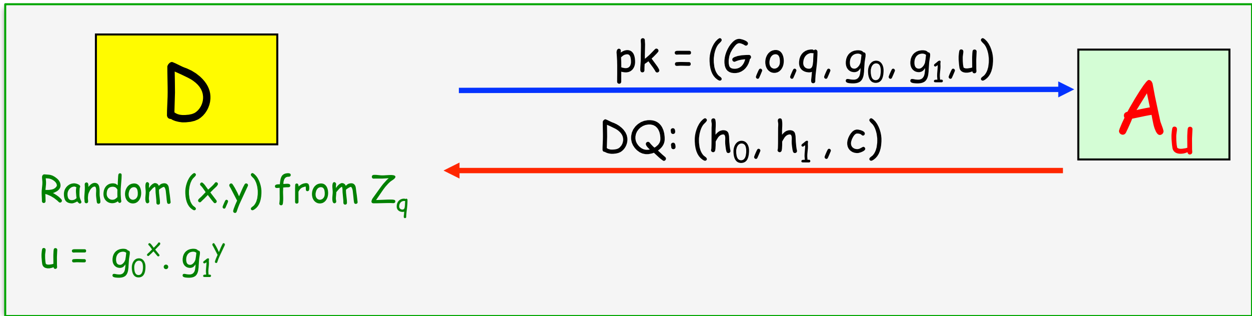
$m = c/v$

Claim. Just one decryption query is enough for an unbounded powerful adversary A_u to know x,y and guess b with probability 1.

Proof: A_u can compute discrete log of u & g_1, say R & α

$$u = g_0^R = (g_0^x . g_1^y) = g_0^{x + \alpha y} \longrightarrow x + \alpha y = R \text{ --- } (1)$$

A_u need another (linearly) independent equation on x and y to recover them. Can Decryption Query help?



Random (x,y) from Z_q

$u = g_0^x . g_1^y$

Is the Scheme CCA1 secure?

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

pk= (G, o, q, g_0, g_1, u), sk = (x,y)

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1, c)

Dec$_{sk = (x,y)}(h_0, h_1, c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Claim. Just one decryption query is enough for an unbounded powerful adversary to know x,y and guess b with probability 1.

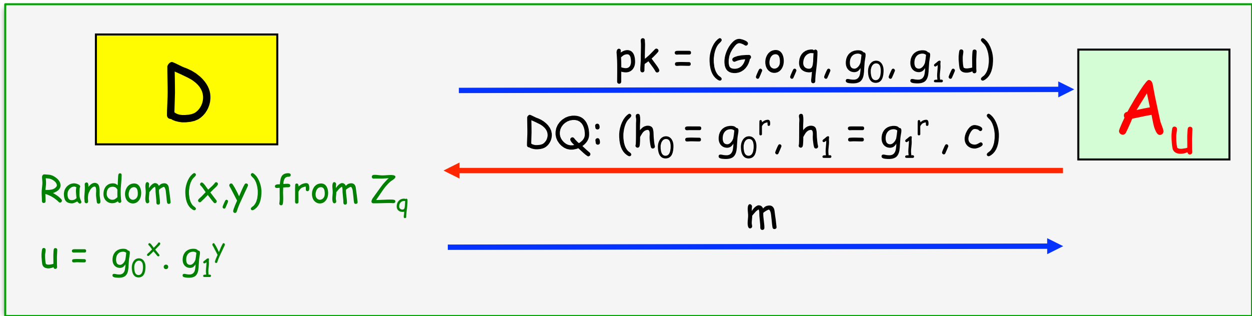
Proof: A_u can compute discrete log of u & g_1, say R & α

$$u = g_0^R = (g_0^x . g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R \quad \text{--- (1)}$$

A_u need another (linearly) independent equation on x and y to recover them. Can Decryption Query help?

$$c/m = v = g_0^{R'} = (h_0^x . h_1^y) = g_0^{rx + r\alpha y} \longrightarrow rx + r\alpha y = R' \quad \text{--- (2)}$$

(linearly) Dependent ☹ No use



Is the Scheme CCA1 secure?

$\mathrm{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G, o, q, g_0, g_1, u)$, $sk = (x,y)$

$\mathrm{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1, c)

$\mathrm{Dec}_{sk = (x,y)}(h_0, h_1, c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Claim. Just one decryption query is enough for an unbounded powerful adversary to know x,y and guess b with probability 1.

Proof: A_u can compute discrete log of u & g_1, say R & α

$$u = g_0^R = (g_0^x . g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R \text{ --- (1)}$$

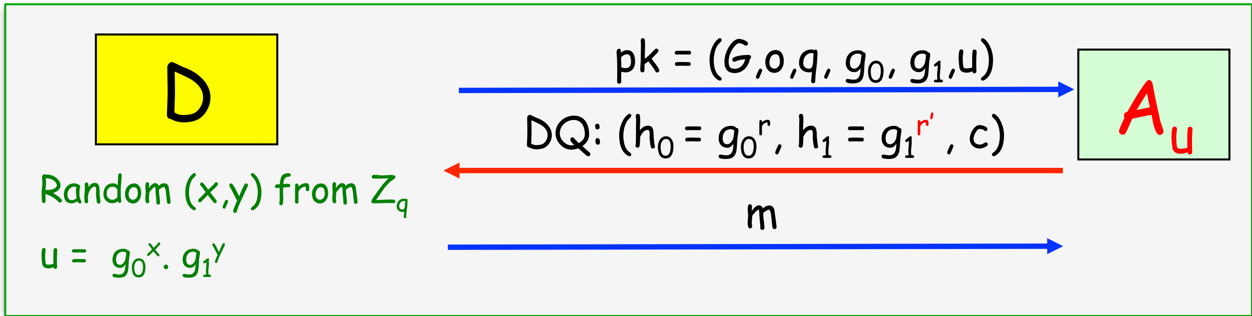
A_u need another (linearly) independent equation on x and y to recover them. Can Decryption Query help?

$$c/m = v = g_0^{R'} = (h_0^x . h_1^y) = g_0^{rx + r'\alpha y} \longrightarrow rx + r'\alpha y = R' \text{ --- (2)}$$

(linearly) independent ☺

Solving (1) & (2) gives the secret key (x,y)

$$\Pr\left(\mathrm{PubK}^{cpa}_{A_u,\overline{\Pi}}(n) = 1 \right) = 1$$



D

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G, o, q, g_0, g_1, u)$

DQ: $(h_0 = g_0^r, h_1 = g_1^{r'}, c)$

m

A_u

CPA Secure Scheme

$Gen(1^n)$

(G, o, q, g_0, g_1)

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$pk = (G,o,q, g_0,g_1, u)$, $sk = (x,y)$

$Enc_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$ (Way2)

(h_0, h_1 , c)

$Dec_{sk = (x,y)}(h_0, h_1 , c)$

$v = h_0^x . h_1^y$ (Way1)

$m = c/v$

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

A, p(n): $Pr\left(PubK^{cpa}_{A,\Pi}(n) = 1\right) > \frac{1}{2} + 1/p(n)$ $Pr\left(PubK^{cpa}_{A,\overline{\Pi}}(n) = 1\right) = \frac{1}{2}$

=

Pr [D(DDH tup

Illegal Decryption Query: $(g_0^r, g_1^{r'} , c)$

A checking mechanism in Dec so that illegal queries will be REJECTED with very high probability

DDH or non-DDH tuple?

$(G,o,q,g_0, g_1, h_0, h_1)$

1 if b = b′

0 otherwise

Random (x,y) from Z_q

$u = g_0^x . g_1^y$

$c = h_0^x . h_1^y . m_b$

$pk = (G,o,q, g_0, g_1,u)$

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h_0, h_1 , c)

$b' \in \{0, 1\}$

CCA1 Scheme

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x,y,x',y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

pk= (G,o,q, g_0,g_1,u,e), sk = (x,y,x',y')

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r$ (Way2)

(h_0, h_1 , c, f)

Dec$_{sk = (x,y,x',y')}(h_0, h_1 , c, f)$

$f = h_0^{x'} h_1^{y'}$ (Way1) ??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Claim. An unbounded powerful adversary computes (x,y) except with neg. probability.
Therefore it can guess bit b with probability no better than ½ + negl(.).

Proof: A_u can compute discrete log of u, e g_1, say R, S & α

$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R$ -(1)

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'} \longrightarrow x' + \alpha y' = S$ -(2)

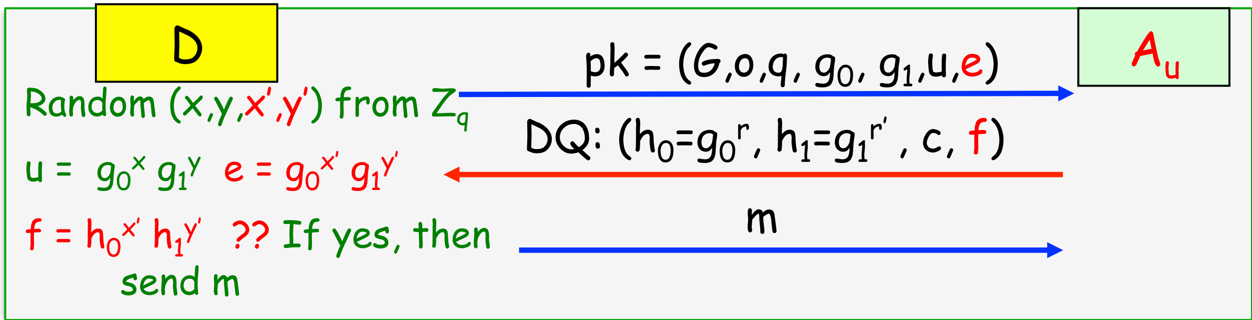
What if A_u can guess f so that $f = h_0^{x'} h_1^{y'}$?? Do you see the disaster???????

$f = g_0^{S'} = (h_0^{x'} h_1^{y'}) = g_0^{rx+r'\alpha y} \longrightarrow rx' + r'\alpha y' = S'$ --- (3)   (linearly) independent of (2)

$c/m = v = g_0^{R'} = (h_0^x . h_1^y) = g_0^{rx+r'\alpha y} \longrightarrow rx + r'\alpha y = R'$ --- (4)   (linearly) independent of (1)

Solving (1) & (4) gives the secret key (x,y)

$$\Pr\left(\mathrm{PubK}^{cpa}_{A_u,\overline{\Pi}}(n) = 1 \right) = 1$$



D

Random (x,y,x',y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$f = h_0^{x'} h_1^{y'}$?? If yes, then send m

A_u

pk = (G,o,q, g_0, g_1,u,e)

DQ: $(h_0=g_0^r, h_1=g_1^{r'} , c, f)$

m

Security Proof of CCA1 Scheme

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

pk= $(G, o, q, g_0, g_1, u, e)$, sk = (x, y, x', y')

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r$ (Way2)

(h_0, h_1, c, f)

Dec$_{sk = (x,y,x',y')}(h_0, h_1, c, f)$

$f = h_0^{x'} h_1^{y'}$ (Way1) ??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Claim. An unbounded powerful adversary computes (x,y) except with neg. probability. Therefore it can guess bit b with probability no better than ½ + negl(.).

Proof: A_u can compute discrete log of u & g_1, say R & α

$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y}$ ⟶

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'}$ ⟶

Yes! It does. A_u knows its chosen value in the first DQ is NOT a possibility. Next time it can guess f from G minus that value

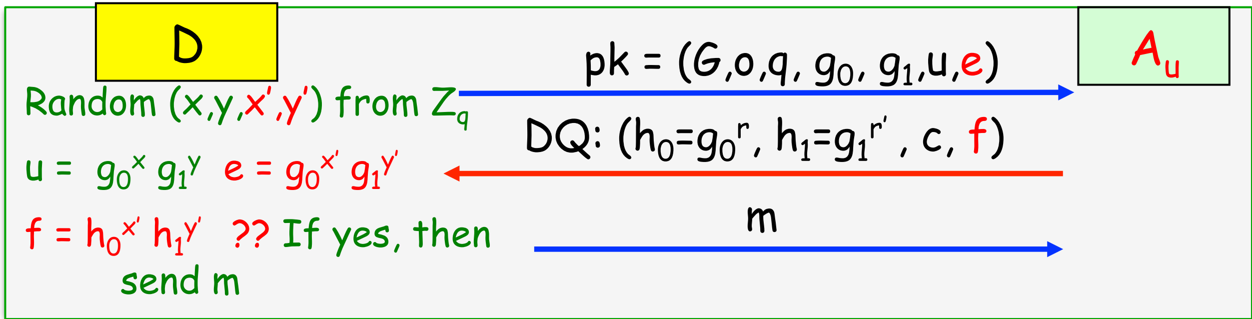
What is the prob of A_u guessing f s

Recall that $h_0^{x'} h_1^{y'}$ is uniformly random for A_u even given $(g_0, g_1, h_0 = g_0^r, h_1 = g_1^{r'}, e)$

Pr[A_u succeeds in first DQ] = 1/|G| --- negligible

But A_u can make polynomials many attempts say, t many.

Does getting rejected in the first DQ help in succeeding second DQ?



D

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$f = h_0^{x'} h_1^{y'}$?? If yes, then send m

A_u

pk = $(G, o, q, g_0, g_1, u, e)$

DQ: $(h_0 = g_0^r, h_1 = g_1^{r'}, c, f)$

m

Security Proof of CCA1 Scheme

$\mathrm{Gen}(1^n)$
(G, o, q, g_0, g_1)
Random (x, y, x', y') from Z_q
$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$
$pk = (G, o, q, g_0, g_1, u, e)$, $sk = (x, y, x', y')$

$\mathrm{Enc}_{pk}(m)$
Random r from Z_q
$h_0 = g_0^r$, $h_1 = g_1^r$
$c = u^r . m = v.m$; $f = e^r$ (Way2)
(h_0, h_1, c, f)

$\mathrm{Dec}_{sk = (x,y,x',y')}(h_0, h_1, c, f)$
$f = h_0^{x'} h_1^{y'}$ (Way1) ??
$v = h_0^x h_1^y$ (Way1)
$m = c/v$

Claim. An unbounded powerful adversary computes (x,y) except with neg. probability. Therefore it can guess bit b with probability no better than ½ + negl(.).

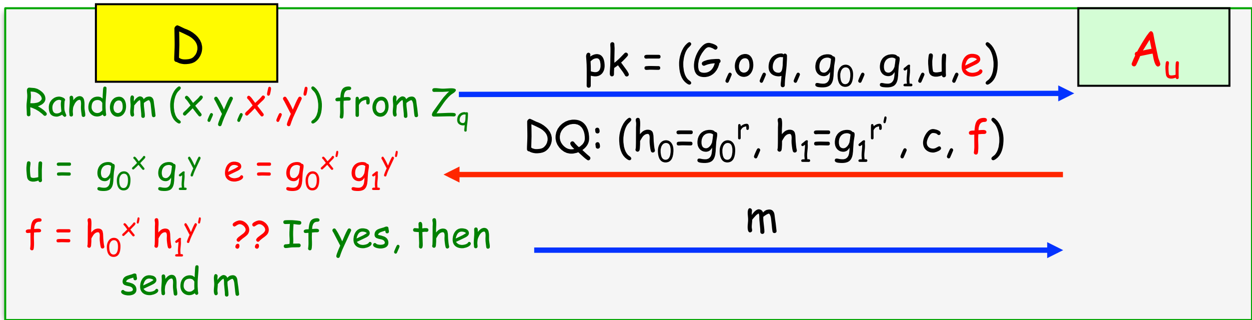
Proof: A_u can compute discrete log of u & g_1, say R & α

$u = g_0^R = (g_0^x g_1^y) = g_0^{x + \alpha y} \longrightarrow x + \alpha y = R$ --- (1)

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x' + \alpha y'} \longrightarrow x' + \alpha y' = S$ --- (2)

What is the prob of A_u guessing f so that $f = h_0^{x'} h_1^{y'} = g_0^{rx + \alpha r' y}$ in his SECOND DQ

Pr[A_u succeeds in second DQ] = 1 / (|G|-1) - negligible



D

Random (x, y, x', y') from Z_q
$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$
$f = h_0^{x'} h_1^{y'}$?? If yes, then send m

Security Proof of CCA1 Scheme

$\text{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$pk = (G, o, q, g_0, g_1, u, e)$, $sk = (x, y, x', y')$

$\text{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r .m = v.m$; $f = e^r$ (Way2)

(h_0, h_1, c, f)

$\text{Dec}_{sk = (x,y,x',y')}(h_0, h_1, c, f)$

$f = h_0^{x'} h_1^{y'}$ (Way1) ??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Claim. An unbounded powerful adversary computes (x,y) except with neg. probability. Therefore it can guess bit b with probability no better than ½ + negl(.).

Proof: A_u can compute discrete log of u & g_1, say R & α

$$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R \text{ --- (1)}$$

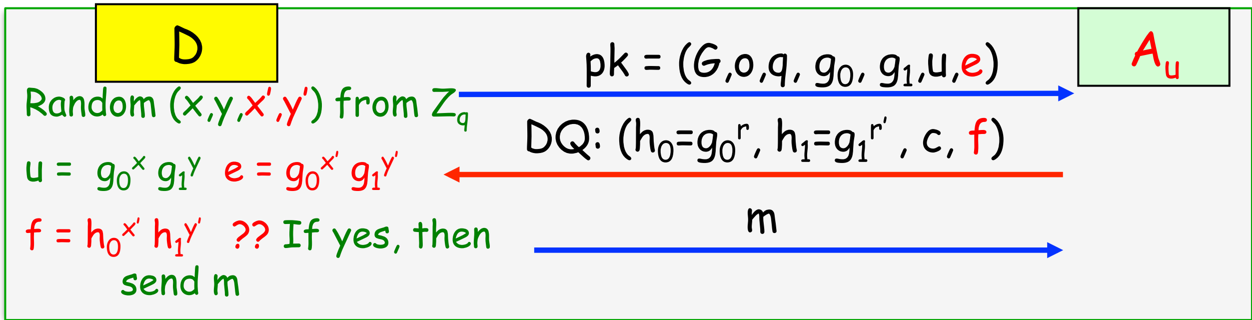
$$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'} \longrightarrow x' + \alpha y' = S \text{ --- (2)}$$

What is the prob of A_u guessing f so that $f = h_0^{x'} h_1^{y'} = g_0^{rx' + \alpha r' y'}$ in his t^{th} DQ (t is the upper bound on the number of DQs)

$\Pr[A_u$ succeeds in t^{th} DQ$] = 1 / (|G|-t)$ - negligible

$\Pr[A_u$ succeeds in one of t DQs$]$
$\leq t / (|G|-t)$ - negligible

$$\Pr\left[\text{PubK}^{cpa}_{A_u,\bar{\Pi}}(n) = 1\right] \leq \tfrac{1}{2} + \text{negl}(.)$$



D

A_u

$pk = (G, o, q, g_0, g_1, u, e)$

Random (x, y, x', y') from Z_q

DQ: $(h_0 = g_0^r, h_1 = g_1^{r'}, c, f)$

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$f = h_0^{x'} h_1^{y'}$?? If yes, then send m

m

Is the Scheme CCA-secure?

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

pk= $(G, o, q, g_0, g_1, u, e)$, sk = (x, y, x', y')

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r$ (Way2)

(h_0, h_1, c, f)

Dec$_{sk = (x,y,x',y')}(h_0, h_1, c, f)$

$f = h_0^{x'} h_1^{y'}$ (Way1) ??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Claim. Just one DQ in post-challenge phase is enough for an unbounded powerful adversary to compute (x,y) completely and guess bit b with probability 1.

Proof: A_u can compute discrete log of u & g_1, say R & α

$$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R \text{ --- (1)}$$

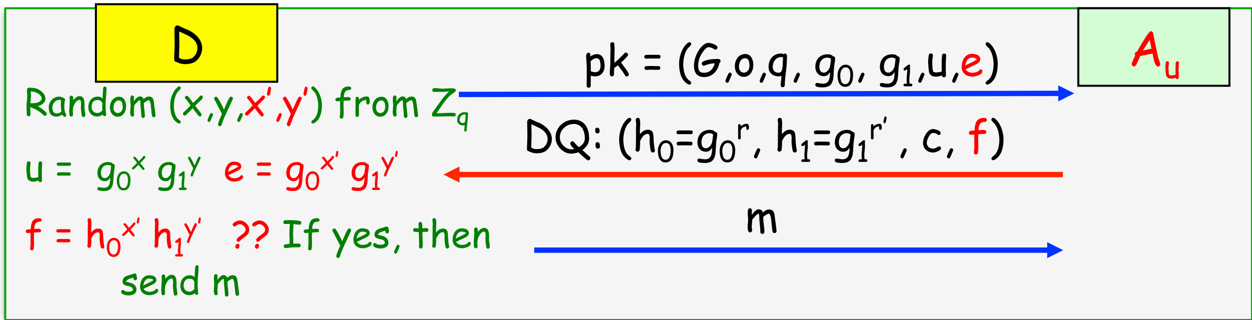
$$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'} \longrightarrow x' + \alpha y' = S \text{ --- (2)}$$

What is the prob of A_u guessing f so that $f = h_0^{x'} h_1^{y'} = g_0^{rx+\alpha r'y}$ in his t^{th} DQ (t is the upper bound on the number of DQs)

Pr[A_u succeeds in t^{th} DQ] = 1 / (|G|-t) - negligible

Pr[A_u succeeds in one of t DQs] ≤ t / (|G|-t) - negligible

$$\Pr\left[\mathrm{PubK}^{\mathrm{cpa}}_{A_u,\overline{\Pi}}(n) = 1\right] \le \frac{1}{2} + \mathrm{negl}(.)$$



D

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$f = h_0^{x'} h_1^{y'}$?? If yes, then send m

Is the Scheme CCA-secure?

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x,y,x',y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

pk= (G,o,q, g_0,g_1,u,e), sk = (x,y,x',y')

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r$ (Way2)

(h_0, h_1, c, f)

Dec$_{sk = (x,y,x',y')}(h_0, h_1, c, f)$

$f = h_0^{x'} h_1^{y'}$ (Way1) ??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Claim. Just one DQ in post-challenge phase is enough for an unbounded powerful adversary to compute (x,y) completely and guess bit b with probability 1.

Proof: A_u can compute discrete log of u & g_1, say R & α

cpa

$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y}$ →

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'}$ →

We need to ensure A_u can not make illegal DQ and get DO service even after seeing the challenge ciphertext.

We are now considering the case wh

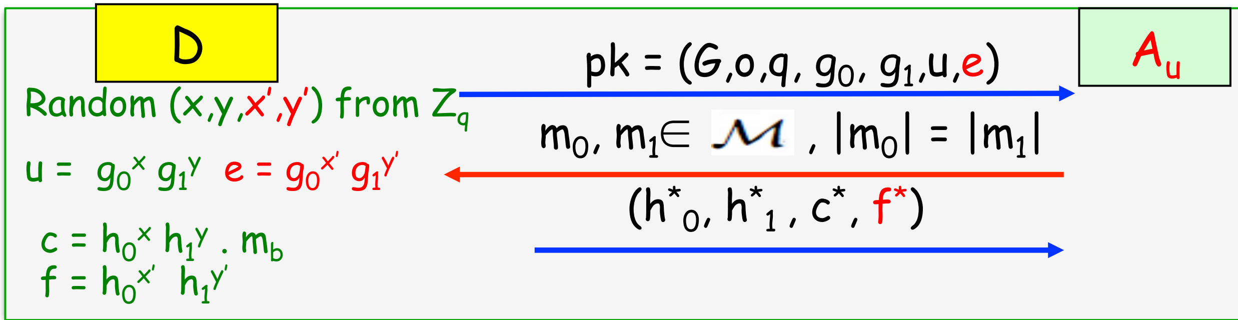
Increase the no. of variables???

$f^* = g_0^{S^*} = (h_0^{x'} h_1^{y'}) = g_0^{rx+r'\alpha y}$ → $rx' + r'\alpha y' = S^*$ --- (3)

(linearly) independent of (2)

Solving (2) & (3) gives (x',y')

Now A_u can make illegal DQ in post-challenge phase and still pass the verification and get m and discover (x,y)



D

A_u

Random (x,y,x',y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$c = h_0^x h_1^y . m_b$

$f = h_0^{x'} h_1^{y'}$

pk = (G,o,q, g_0, g_1,u,e)

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h^*_0, h^*_1, c^*, f^*)

Is the Scheme CCA-secure?

$Gen(1^n)$

(G, o, q, g_0, g_1)

Random (x, y, x', y', x'', y'') from Z_q

$u = g_0^x g_1^y$ $\quad e = g_0^{x'} g_1^{y'}$ $\quad k = g_0^{x''} g_1^{y''}$

pk= $(G, o, q, g_0, g_1, u, e, k)$, sk = (x, y, x', y', x'', y'')

$Enc_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r$; $l = k^r$ (Way2)

(h_0, h_1, c, f, l)

$Dec_{sk = (x,y,x',y',,x'',y'')}(h_0, h_1, c, f, l)$

$f = h_0^{x'} h_1^{y'}$ (Way1) ??

$l = h_0^{x''} h_1^{y''}$ (Way1) ??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Does above help?

Proof: A_u can compute discrete log of u & g_1, say R & α

$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R$ - (1)

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'} \longrightarrow x' + \alpha y' = S$ - (2)

$k = g_0^T = (g_0^{x''} g_1^{y''}) = g_0^{x''+\alpha y''} \longrightarrow x'' + \alpha y'' = T$ - (3)

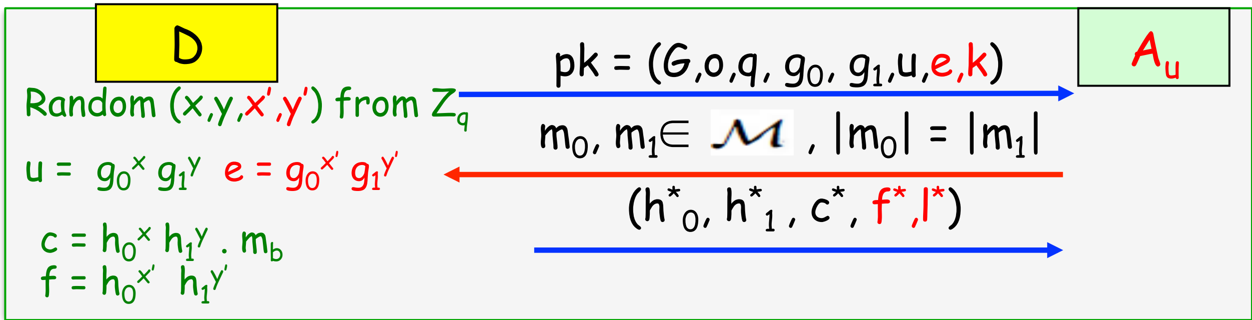
$f^* = g_0^{S^*} = (h_0^{x'} h_1^{y'}) = g_0^{rx'+r'\alpha y'} \longrightarrow rx' + r'\alpha y' = S^*$ - (4)

$l^* = g_0^{T^*} = (h_0^{x''} h_1^{y''}) = g_0^{rx''+r'\alpha y''} \longrightarrow rx'' + r'\alpha y'' = T^*$ - (5)

We are now considering the case when D received a non-DDH tuple (g_0, g_1, h^*_0, h^*_1) and so $h^*_0 = g_0^r$, $h^*_1 = g_1^{r'}$

Now A_u can make illegal DQ in post-challenge phase and still pass the verification and get m and discover (x,y)

Adding more variable in the above way does not help.



D

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $\quad e = g_0^{x'} g_1^{y'}$

$c = h_0^x h_1^y . m_b$

$f = h_0^{x'} h_1^{y'}$

pk = $(G, o, q, g_0, g_1, u, e, k)$

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

$(h^*_0, h^*_1, c^*, f^*, l^*)$

A_u

Is the Scheme CCA-secure?

$\mathrm{Gen}(1^n)$

(G, o, q, g_0, g_1)

Random (x, y, x', y', x'', y'') from Z_q

$u = g_0^x g_1^y \quad e = g_0^{x'} g_1^{y'} \quad k = g_0^{x''} g_1^{y''}$

$pk = (G, o, q, g_0, g_1, u, e, k)$, $sk = (x, y, x', y', x'', y'')$

$\mathrm{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r k^r$ (Way2)

(h_0, h_1, c, f)

$\mathrm{Dec}_{sk = (x,y,x',y',x'',y'')}(h_0, h_1, c, f)$

$f = h_0^{x'+x''} h_1^{y'+y''}$??

$v = h_0^x h_1^y$ (Way1)

$m = c/v$

Does above help?

Proof:

$u = g_0^R = (g_0^x g_1^y) = g_0^{x+\alpha y} \longrightarrow x + \alpha y = R \quad - (1)$

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x'+\alpha y'} \longrightarrow x' + \alpha y' = S \quad - (2)$

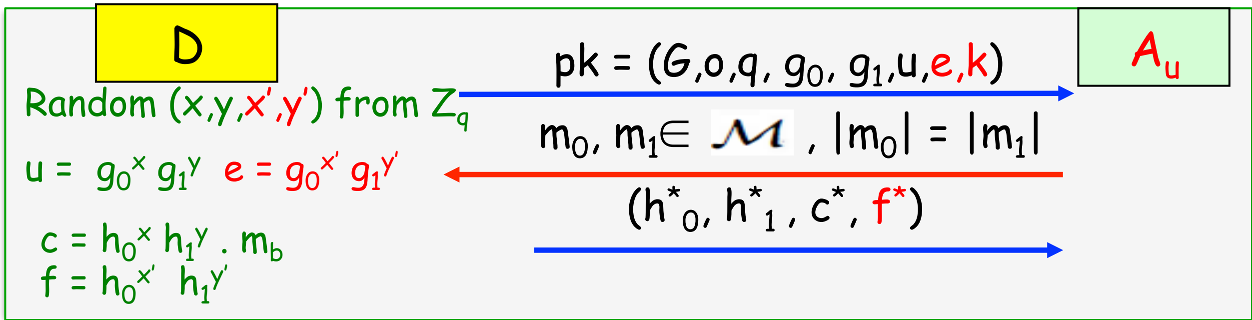
$k = g_0^T = (g_0^{x''} g_1^{y''}) = g_0^{x''+\alpha y''} \longrightarrow x'' + \alpha y'' = T \quad - (3)$

$f^* = g_0^{S^*} = (h_0^{x'+x''} h_1^{y'+y''}) = g_0^{r(x'+x'')+r'\alpha(y'+y'')} \longrightarrow$

$r(x' + x'') + r'\alpha(y' + y'') = S^* \quad - (4)$

From (2), (3) & (4), A_u can compute $(x' + x'')$ and $(y' + y'')$ and that's enough to make an illegal DQ in post-challenge phase and still pass the verification and get m and discover (x,y).

Adding more variable in the above way does not help.



D

Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y \quad e = g_0^{x'} g_1^{y'}$

$c = h_0^x h_1^y . m_b$

$f = h_0^{x'} h_1^{y'}$

A_u

$pk = (G, o, q, g_0, g_1, u, e, k)$

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h^*_0, h^*_1, c^*, f^*)

Is the Scheme CCA-secure?

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x, y, x', y', x'', y'') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$ $k = g_0^{x''} g_1^{y''}$

pk= $(G, o, q, g_0, g_1, u, e, k, H)$, sk = (x, y, x', y', x'', y'')

Enc$_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r k^{\beta r}$

$\beta = H(h_0, h_1, c)$

(h_0, h_1, c, f)

Dec$_{sk = (x,y,x',y',x'',y'')}(h_0, h_1, c, f)$

$\beta = H(h_0, h_1, c)$

$f = h_0^{x' + \beta x''} h_1^{y' + \beta y''}$??

$v = h_0^x h_1^y$

$m = c/v$

Does above help?

Proof:

$u = g_0^R = (g_0^x g_1^y) = g_0^{x + \alpha y} \longrightarrow x + \alpha y = R$ - (1)

$e = g_0^S = (g_0^{x'} g_1^{y'}) = g_0^{x' + \alpha y'} \longrightarrow x' + \alpha y' = S$ - (2)

$k = g_0^T = (g_0^{x''} g_1^{y''}) = g_0^{x'' + \alpha y''} \longrightarrow x'' + \alpha y'' = T$ - (3)

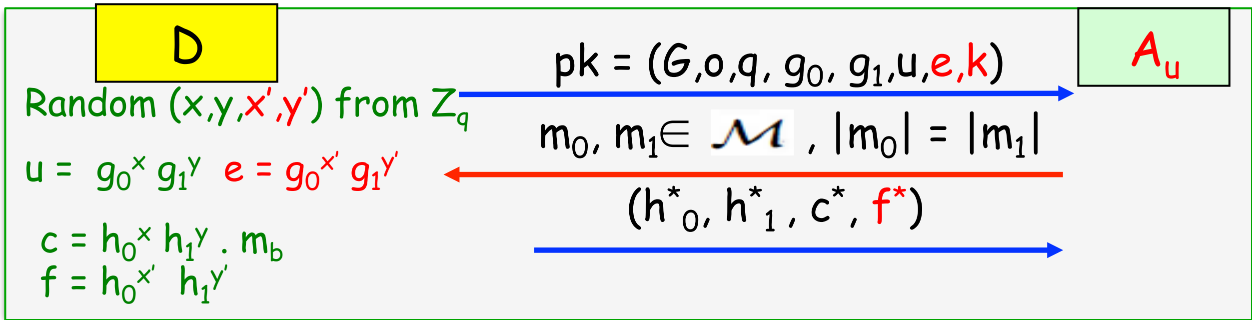
$f^* = g_0^{S'} = (h_0^{x' + \beta x''} h_1^{y' + \beta y''}) = g_0^{r(x' + \beta x'') + r'\alpha(y' + \beta y'')} \longrightarrow$

$r(x' + \beta x'') + r'\alpha(y' + \beta y'') = S'$ - (4)

From (2), (3) & (4), A_u can compute $(x' + \beta x'')$ and $(y' + \beta y'')$ BUT……

…….to make an illegal DQ in post-challenge phase A_u must find a collision for H i.e h'_0, h'_1, c' such that $\beta = H(h'_0, h'_1, c')$ because…..

…….he is not allowed to submit the challenge ciphertext to DO



Random (x, y, x', y') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$

$c = h_0^x h_1^y . m_b$

$f = h_0^{x'} h_1^{y'}$

$pk = (G, o, q, g_0, g_1, u, e, k)$

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h^*_0, h^*_1, c^*, f^*)

The Cramer-Shoup Cryptosystem

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x, y, x', y', x'', y'') from Z_q

$u = g_0^{x} g_1^{y}$ $e = g_0^{x'} g_1^{y'}$ $k = g_0^{x''} g_1^{y''}$

pk= $(G,o,q, g_0,g_1,u,e,k, H)$, sk = (x,y,x',y',x'',y'')

Enc$_{pk}$(m)

Random r from Z_q

$h_0 = g_0^{r}$, $h_1 = g_1^{r}$

$c = u^r .m = v.m$; $f = e^r k^{\beta r}$

$\beta = H(h_0, h_1, c)$

(h_0, h_1, c, f)

Dec$_{sk = (x,y,x',y',x'',y'')}(h_0,h_1,c,f)$

$\beta = H(h_0, h_1, c)$

$f = h_0^{x' + \beta x''} h_1^{y' + \beta y''}$??

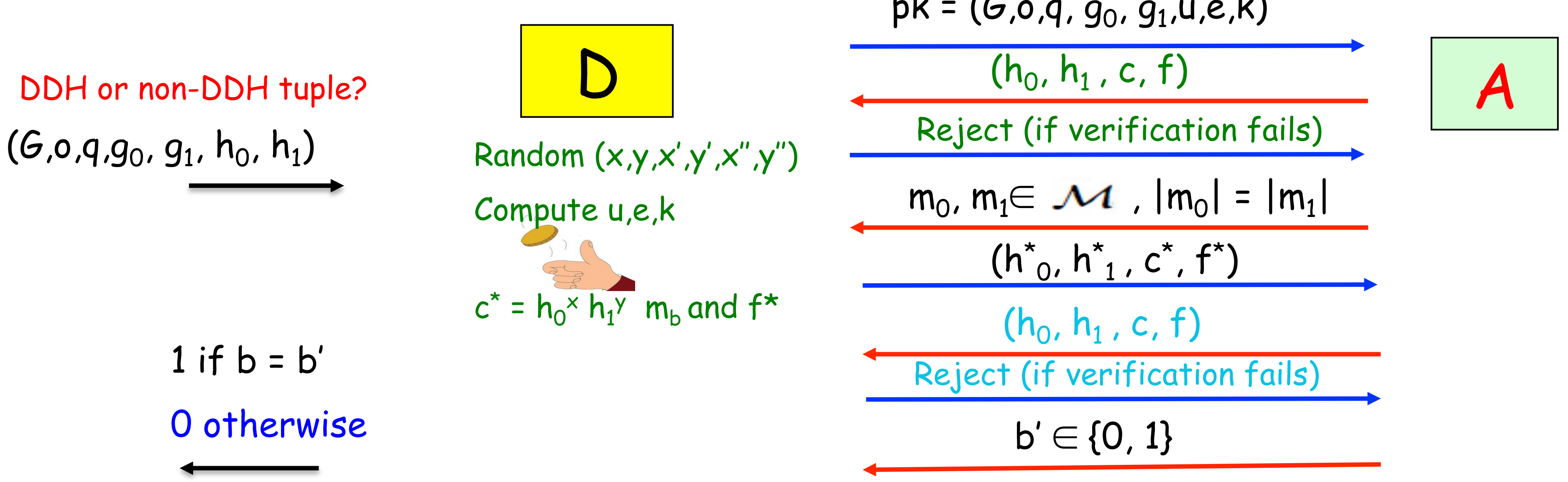
$v = h_0^{x} h_1^{y}$

$m = c/v$

Theorem. If DDH is hard + H is CR HF, then Π is a CCA-secure scheme.

Case I: If $(h_0, h_1, c) = (h^*_0, h^*_1, c^*)$ and $f^* \neq f$ → D will reject

Case II: If $(h_0, h_1, c) \neq (h^*_0, h^*_1, c^*)$ and $f^* = f$ [i.e. $H(h_0, h_1, c) = H(h^*_0, h^*_1, c^*)$] → A has found collision but since H is CR, this happens with negl probability



DDH or non-DDH tuple?

$(G,o,q,g_0, g_1, h_0, h_1)$

1 if b = b'

0 otherwise

D

Random (x,y,x',y',x'',y'')

Compute u,e,k

$c^* = h_0^{x} h_1^{y}$ m_b and f^*

pk = (G,o,q, g_0, g_1,u,e,k)

(h_0, h_1, c, f)

Reject (if verification fails)

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(h^*_0, h^*_1, c^*, f^*)

(h_0, h_1, c, f)

Reject (if verification fails)

$b' \in \{0, 1\}$

A

The Cramer-Shoup Cryptosystem

Gen(1^n)

(G, o, q, g_0, g_1)

Random (x, y, x', y', x'', y'') from Z_q

$u = g_0^x g_1^y$ $e = g_0^{x'} g_1^{y'}$ $k = g_0^{x''} g_1^{y''}$

pk= $(G, o, q, g_0, g_1, u, e, k, H)$, sk = (x, y, x', y', x'', y'')

$\text{Enc}_{pk}(m)$

Random r from Z_q

$h_0 = g_0^r$, $h_1 = g_1^r$

$c = u^r.m = v.m$; $f = e^r k^{\beta r}$

$\beta = H(h_0, h_1, c)$

(h_0, h_1, c, f)

$\text{Dec}_{sk = (x, y, x', y', x'', y'')}(h_0, h_1, c, f)$

$\beta = H(h_0, h_1, c)$

$f = h_0^{x' + \beta x''} h_1^{y' + \beta y''}$??

$v = h_0^x h_1^y$

$m = c/v$

Theorem. If DDH is hard + H is CR HF, then Π is a CCA-secure scheme.

Case III: $H(h_0, h_1, c) \neq H(h^*_0, h^*_1, c^*)$]: There is a possibility that it is a valid ciphertext. But it can happen by sheer luck.

We can have four INDEPENDENT constraints on x′,y′.x″.y″ : (1) e (2) k (3) challenge cipher (4) DO => Breaking security

But before making DO query A had only three constraints and so finding an matching f for the DO can be donewith prob at most 1/|G|

DDH or non-DDH tuple?

$(G, o, q, g_0, g_1, h_0, h_1)$ →

D

Random (x, y, x', y', x'', y'')

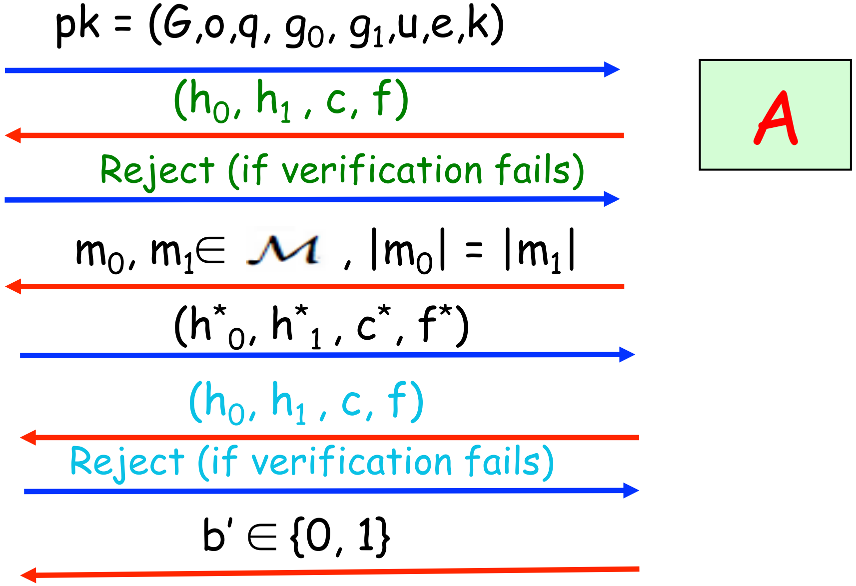
Compute u,e,k

$c^* = h_0^x h_1^y$ m_b and f*

1 if b = b′

0 otherwise

←



$pk = (G, o, q, g_0, g_1, u, e, k)$ →

(h_0, h_1, c, f) ←

Reject (if verification fails) →

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$ ←

(h^*_0, h^*_1, c^*, f^*) →

(h_0, h_1, c, f) ←

Reject (if verification fails) →

$b' \in \{0, 1\}$ ←

A

Public Key Summary

Primitives	Security Notions	Assumptions
PKE KEM Hybrid Encryption	CPA CCA Adaptive Attack (Non-committing Encryption) Selective Opening Attack (Deniable Encryption)	>> Close Relatives of DL assumptions- CDH, DDH, HDH, ODH >> RSA Assumption (Padded RSA, RSA OAEP) >> Factoring assumptions (Rabin Cryptosystem) >> Quadratic Residuacity Assumptions (Micali-Goldwasser) >> Decisional Composite Residuacity (DCR) Assumptions (Paillier) >> Lattice-based Assumptions LWE, LPN (Regev) Many more assumptions

Thank You!